

INSTRUCTION NUMBER EXPANSION METHOD IN PARALLEL PROCESSOR, AND PARALLEL PROCESSORS

Publication number: JP10011289

Publication date: 1998-01-16

Inventor: KONDO YOSHIKAZU

Applicant: MITSUBISHI ELECTRIC CORP

Classification:

- international: G06F9/30; G06F9/318; G06F9/38; G06F9/30;
G06F9/318; G06F9/38; (IPC1-7): G06F9/38; G06F9/30

- European: G06F9/30T; G06F9/318; G06F9/38T2

Application number: JP19960158505 19960619

Priority number(s): JP19960158505 19960619

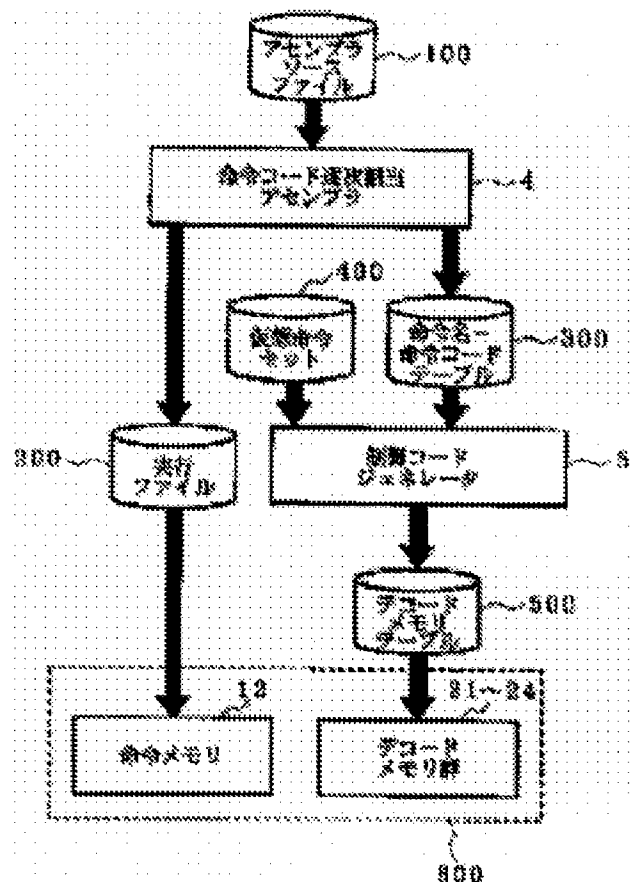
Also published as:

US5848255 (A1)

Report a data error here

Abstract of JP10011289

PROBLEM TO BE SOLVED: To practically reduce instruction extension, to mitigate the limit of an I/O band width and to increase functions mountable to one substrate by respectively writing plural control codes corresponding to plural instructions to plural instruction decoders based on decoding information prepared at the time of assembling a source file. **SOLUTION:** An instruction code successive allocation assembler 4 outputs which instruction has been allocated to which instruction, to an instruction name-instruction code correspondence table 300 for indicating the correspondence relation of an instruction name and the instruction code. A control code generator 5 finds the control code corresponding to the instruction code based on the instruction name-instruction code correspondence table 300 and a virtual instruction set 400 for indicating the correspondence relation of the instructions and all the control codes physically realizable on the parallel processor 600 and outputs it to a decoding memory table 500. Then, the contents of the decoding memory table 500 are transferred to a decoding memory group (instruction decoder) prior to the activation of the instruction.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-11289

(43)公開日 平成10年(1998)1月16日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/38	3 7 0		G 0 6 F 9/38	3 7 0 X
9/30	3 1 0		9/30	3 1 0 C
				3 1 0 E

審査請求 未請求 請求項の数12 O L (全 11 頁)

(21)出願番号 特願平8-158505

(22)出願日 平成8年(1996)6月19日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 近藤 由和

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

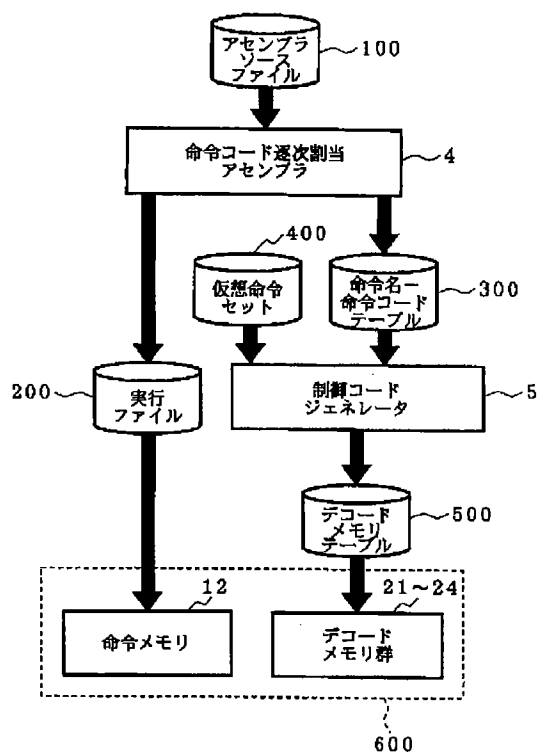
(74)代理人 弁理士 田澤 博昭 (外2名)

(54)【発明の名称】 並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサ

(57)【要約】

【課題】 限られた命令語長を使う限り命令数を増やすことが困難であった。

【解決手段】 命令解読器としての書き換え可能なデコードメモリ群21～24を設け、アセンブル時に、所定の複数の命令コードと複数の制御コードとの対応関係を示すデコードメモリテーブル500を実行ファイル200の作成と同時に生成し、これに従って上記の複数の制御コードをデコードメモリ群21～24に書き込む。



【特許請求の範囲】

【請求項1】 書き換え可能な素子で構成された複数の命令解読器を設け、ソースファイルのアセンブル時に作成したデコード情報に基づいて複数の命令に対応した複数の制御コードを前記複数の命令解読器にそれぞれ書き込むことにより複数の演算部を用いて前記複数の命令を実行する並列処理プロセッサにおける命令数拡張方法。

【請求項2】 複数の命令と所定の複数の命令コードとの対応関係を示す命令名-命令コード対応テーブルを作成し、前記複数の命令と複数の制御コードとの対応関係を示す仮想命令セットと前記命令名-命令コード対応テーブルとを参照し、前記複数の命令コードと前記複数の制御コードとを対応させたデコード情報としてのデコードメモリテーブルを作成し、前記複数の制御コードを前記複数の命令解読器にそれぞれ書き込むことを特徴とする請求項1記載の並列処理プロセッサにおける命令数拡張方法。

【請求項3】 所定の複数の命令コードのそれぞれをアドレスとして設定することにより、複数の制御コードを複数の命令解読器にそれぞれ書き込むことを特徴とする請求項2記載の並列処理プロセッサにおける命令数拡張方法。

【請求項4】 複数の命令解読器に書き込まれた制御コードを再設定するための再設定命令を設け、前記再設定命令を実行した場合には、予め作成され、複数の命令コードと他の複数の制御コードとの対応関係を示す他のデコードメモリテーブルに基づき前記他の複数の制御コードを複数の命令解読器にそれぞれ書き込むことを特徴とする請求項2または請求項3記載の並列処理プロセッサにおける命令数拡張方法。

【請求項5】 複数の命令解読器に書き込まれた制御コードを書き換えるための書き換え命令を設け、前記書き換え命令を実行した場合には、予め並列処理プロセッサ内に記憶された複数の制御コードを前記複数の命令解読器にそれぞれ書き込むことを特徴とする請求項1から請求項3のうちのいずれか1項記載の並列処理プロセッサにおける命令数拡張方法。

【請求項6】 並列処理プロセッサの初期化の際に、複数の命令解読器の少なくとも一部に一つまたは複数の所定の制御コードを書き込むことを特徴とする請求項1から請求項5のうちのいずれか1項記載の並列処理プロセッサにおける命令数拡張方法。

【請求項7】 所定の制御コードは出現頻度が高い命令に関するものであることを特徴とする請求項6記載の並列処理プロセッサにおける命令数拡張方法。

【請求項8】 書き換え可能な素子で構成された複数の命令解読器と、外部から前記複数の命令解読器のいずれかを選択するために入力されるアドレス信号を受信する第1の入力ポートと、デコード情報に基づき外部から前記複数の命令解読器に制御コードをそれぞれ書き込むた

めに前記制御コードを受信する第2の入力ポートとを備えた並列処理プロセッサ。

【請求項9】 複数の命令解読器は、第1の入力ポートを介してアドレス信号を受信するためのアドレスポートと、第2の入力ポートを介して制御コードを受信するための複数のデータ入力ポートと、対応する演算器に制御コードを送信するための複数のデータ出力ポートとを備えたランダムアクセスメモリであることを特徴とする請求項8記載の並列処理プロセッサ。

【請求項10】 複数の命令解読器を再設定すべく書き換えるために再設定要求を外部に出力するための再設定レジスタを備えたことを特徴とする請求項8または請求項9記載の並列処理プロセッサ。

【請求項11】 所定の書き換え命令に応じて複数の命令解読器のいずれか一つを選択するためのアドレスを保持する命令選択レジスタと、書き換え命令に基づき予め保持していた制御コードを前記命令選択レジスタで指定される前記複数の命令解読器の一つに書き込むために一旦記憶する命令定義レジスタとを備えたことを特徴とする請求項8または請求項9記載の並列処理プロセッサ。

【請求項12】 並列処理プロセッサの初期化の際に複数の命令解読器の少なくとも一部に一つまたは複数の所定の制御コードを書き込むために、所定の制御コードを記憶するための読み出し専用メモリを備えたことを特徴とする請求項8から請求項11のうちのいずれか1項記載の並列処理プロセッサ。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】この発明は、画像処理、科学技術計算、ニューラルネットワーク等の多様なアプリケーションを効率よく実行するための並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサに関するものである。

【0002】

【従来の技術】図7は、例えば中野ひらく等による“80-MFLOPS (ピーク) 64ビット並列コンピュータ用マイクロプロセッサ (An 80-MFLOPS (Peak) 64-b Microprocessor for Parallel Computer”、アイ・イー・イー・イー ジャーナル オブ ソリッドステート サーキット (IEEE JOURNAL OF SOLID-STATE CIRCUITS)、Vol. 12, No. 3, 365~372ページ、1992年に開示された従来の並列処理プロセッサを示すブロック図であり、図において、10はパイプライン化された制御部、11はプログラムカウンタ、12は命令メモリ（あるいは命令キャッシュ）、13は命令レジスタ、14は命令デコード、15~18は制御コードレジスタ、50~53は制御部10に並列に接続された第1から第4の演算部である。

【0003】次に動作について説明する。命令デコーダ14は、プログラムカウンタ11に格納されたアドレスに基づいて命令レジスタ13に保持された命令コード（命令メモリ12に含まれる内容）を、第1から第4の演算部50～53に供給する制御コードへ伸長する。命令デコーダ14は書き換え不可能な素子で構成されており、命令コードと制御コードとの対応関係は並列処理プロセッサのチップ製造時に決定される。

【0004】近年、LSI技術が進展し、単一シリコン基板上に多数の演算器を集積した並列処理プロセッサの開発が可能になってきた。それに伴い、演算器の利用率を高めるための種々のプロセッサ制御方法が開発された。例えば、演算器利用率が高くなるよう動的に命令を発行するスーパースカラ技術、複数の演算器の個々の命令を1つの長い命令語上に配置し、コンパイル時に静的に命令割当を行なうVLIW (Very Long Instruction Word) 技術などがあげられる。他方、プロセッサ利用技術も多様化し、DSPに見られるような多数の命令（並列転送機能、条件参照実行、アドレス計算付き実行、ワード分割並列実行）が必要になってきている。

【0005】このような流れは、次世代の並列処理プロセッサに次のような問題を引き起こすことが予想される。

【0006】(1) 命令語長の増大。

多数の演算器の利用率を高めるためには、VLIW技術に見られるように命令語長を長くし多数の命令を割り当てる必要がある。しかし、この方法には語長に見合ったI/Oバンド幅の増加が同時に行なわれないと性能が出ないという欠点がある。既に現状の並列処理プロセッサのI/O転送速度は飽和しつつあり、命令語長を劇的に増やすことは困難である。

【0007】(2) I/Oピン数制限の問題。

(1)の問題を補うために、I/Oピン数を増やすことで必要なI/Oバンド幅を得ることも考えられる。ただし、この方法も出力バッファの同時スイッチング雑音により制約を受け、さらに、電源供給に伴う電源ピン数増加のために100～300ピン程度が限界である。

【0008】(3) 命令出現頻度の偏り。

多数の命令を搭載することで別の問題が持ち上がる。それは、命令を増やせば増やすほど、個々の命令の出現率が低下し、殆ど使われなくなる命令がでてくることである。図8は256個の命令を実行できる24個の演算器を備えたSIMD (Single-Instruction Multiple-Data) 型並列処理プロセッサにおける命令出現頻度の実測値である（数値演算用および画像処理システム用ライブラリ関数に含まれる命令を分析した）。この図から、全命令の約半分が0.01%以下の出現頻度であることがわかる。つまり、めったに現れない命令のために命令表を浪費していることに

なる。他方、これらの低出現頻度命令は、並列転送、条件参照実行、ワード分割実行、アドレス計算付き実行等の並列度の高い演算を多く含み、命令全体として速度向上に寄与しているので、簡単には廃止できない。

【0009】

【発明が解決しようとする課題】従来の並列処理プロセッサは以上のように構成されているので、命令コードと制御コードとの対応関係は設計時に決める必要があり、再定義することはできないという課題があった。

【0010】また、複数の演算部を自由に組合せた動作を定義したり、複数の演算部に対し新しい機能を追加するためには、命令語長（命令メモリ12の語長）を大きくする必要があるが、上記したように命令語長の増大が困難であるという課題があった。

【0011】従って、現実の限られた命令語長を使う限り、命令数を増やすことも、多数の演算器を効率よく制御することも困難であるという課題があった。

【0012】この発明は上記のような課題を解決するためになされたもので、限られた命令語長で複数の機能（命令）を実現することができる並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサを得ることを目的とする。

【0013】また、この発明は、実質的に命令語長を減らすことでI/Oバンド幅の制限を緩和し、1つのシリコン基板上に搭載できる機能を増やすことができる並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサを得ることを目的とする。

【0014】

【課題を解決するための手段】請求項1記載の発明に係る並列処理プロセッサにおける命令数拡張方法は、書き換え可能な素子で構成された複数の命令解読器を設け、ソースファイルのアセンブル時に作成したデコード情報に基づいて複数の命令に対応した複数の制御コードを複数の命令解読器にそれぞれ書き込むものである。

【0015】請求項2記載の発明に係る並列処理プロセッサにおける命令数拡張方法は、複数の命令と所定の複数の命令コードとの対応関係を示す命令名-命令コード対応テーブルを作成し、複数の命令と複数の制御コードとの対応関係を示す仮想命令セットと命令名-命令コード対応テーブルとを参照し、複数の命令コードと複数の制御コードとを対応させたデコード情報としてのデコードメモリテーブルを作成するものである。

【0016】請求項3記載の発明に係る並列処理プロセッサにおける命令数拡張方法は、所定の複数の命令コードのそれぞれをアドレスとして設定するものである。

【0017】請求項4記載の発明に係る並列処理プロセッサにおける命令数拡張方法は、複数の命令解読器に書き込まれた内容を再設定するための再設定命令を設け、再設定命令を実行した場合には、予め作成され、複数の命令コードと他の複数の制御コードとの対応関係を示す

他のデコードメモリテーブルに基づき他の複数の制御コードを複数の命令解読器にそれぞれ書き込むものである。

【0018】請求項5記載の発明に係る並列処理プロセッサにおける命令数拡張方法は、複数の命令解読器に書き込まれた内容を書き換えるための書き換え命令を設け、書き換え命令を実行した場合には、予め並列処理プロセッサ内に記憶された複数の制御コードを複数の命令解読器にそれぞれ書き込むものである。

【0019】請求項6記載の発明に係る並列処理プロセッサにおける命令数拡張方法は、並列処理プロセッサの初期化の際に、複数の命令解読器の少なくとも一部に一つまたは複数の所定の制御コードを書き込むものである。

【0020】請求項7記載の発明に係る並列処理プロセッサにおける命令数拡張方法は、上記の所定の制御コードは出現頻度が高い命令に関するものであるものである。

【0021】請求項8記載の発明に係る並列処理プロセッサは、書き換え可能な素子で構成された複数の命令解読器と、外部から複数の命令解読器のいずれかを選択するために入力されるアドレス信号を受信する第1の入力ポートと、デコード情報に基づき外部から複数の命令解読器に制御コードをそれぞれ書き込むために制御コードを受信する第2の入力ポートとを備えたものである。

【0022】請求項9記載の発明に係る並列処理プロセッサは、複数の命令解読器が、第1の入力ポートを介してアドレス信号を受信するためのアドレスポートと、第2の入力ポートを介して制御コードを受信するための複数のデータ入力ポートと、対応する演算器に制御コードを送信するための複数のデータ出力ポートとを備えたランダムアクセスメモリであるものである。

【0023】請求項10記載の発明に係る並列処理プロセッサは、複数の命令解読器を再設定すべく書き換えるために再設定要求を外部に出力するための再設定レジスタを備えたことを特徴とする並列処理プロセッサであるものである。

【0024】請求項11記載の発明に係る並列処理プロセッサは、所定の書き換え命令に応じて複数の命令解読器のいずれか一つを選択するためのアドレスを保持する命令選択レジスタと、書き換え命令に基づき予め保持していた制御コードを命令選択レジスタで指定される複数の命令解読器の一つに書き込むために一旦記憶する命令定義レジスタとを備えたものである。

【0025】請求項12記載の発明に係る並列処理プロセッサは、並列処理プロセッサの初期化の際に複数の命令解読器の少なくとも一部に一つまたは複数の所定の制御コードを書き込むために、所定の制御コードを記憶するための読み出し専用メモリを備えたものである。

【0026】

【発明の実施の形態】以下、この発明の実施の一形態を説明する。

実施の形態1. 図1はこの発明の実施の形態1による並列処理プロセッサにおける命令数拡張方法を実現する並列処理プロセッサの主要部の構成を示すブロック図であり、図において、図7と同一の部分には同一符号を付して重複説明を省略する。また、図において、20は制御部、21～24はデータ入力ポートとデータ出力ポートとが別々に設けられたランダムアクセスメモリ(RAM)から構成され、命令解読器として設けられたデコードメモリ群、25～28はデコードメモリ群21～24のアドレスポート、29～32はデコードメモリ群21～24のデータ入力ポート、33～36はデコードメモリ群21～24のデータ出力ポートである。

【0027】さらに、37はデコードメモリ群21～24の内容を並列処理プロセッサ600の外部から設定するためのアドレス指定ポート(第1の入力ポート)、38はデコードメモリ群21～24のアドレスポート25～28をアドレス指定ポート37へ接続するためのマルチプレクサ、39はデコードメモリ群21～24へ並列処理プロセッサ600の外部からデータを書き込むためのデータ書き込みポート(第2の入力ポート)、60は異なるデコードメモリテーブル(命令コードと制御コードとの対応関係を示す対応テーブル、即ち、デコード情報)が必要となる実行ファイル間の境界において、デコードメモリ群21～24の再設定要求を並列処理プロセッサ600のチップ外部へ発生させる際に、例えば、状態“1”に遷移する再設定レジスタ、61は再設定レジスタ60からの再設定要求を外部に送信するための再設定要求出力ポートである。

【0028】図1に示すように、デコードメモリ群21～24のアドレスポート25～28はマルチプレクサ38を介して命令レジスタ13に接続され、データ出力ポート33～36はそれぞれ制御コードレジスタ15～18に接続されている。これにより、後で述べるように、命令コードと制御コードとの対応関係が自由に設定できる。

【0029】この発明による並列処理プロセッサの命令数拡張方法および並列処理プロセッサは、アセンブラソースファイル内の命令使用に関する局所性、即ち、多数の命令全体が常に均一に使われるのではなく、プログラムの種類によって使う命令が偏る性質を利用している。例えば、映像の画素単位の処理(コンボリューション等)には浮動小数点命令を使わないし、DSPが採用している特殊なアドレッシングモード(ビット反転等)は信号処理用途以外のコードには現れにくい。この性質は、長い命令フィールドに複数の機能を割り当てるVLIW技術にも当てはまる性質である。つまり、割り当てられる命令の組合せは均一でなくプログラムに依存した偏りが存在すると考えられる。この発明は、このような命令

出現の局所性を利用し、出現頻度の低い命令群の制御コードを同一の命令コードに再定義することによって、少ない語長で多数の命令・機能を仮想的に実現するものである。

【0030】次に動作について説明する。図2はこの実施の形態1による並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサにおけるアセンブラソースファイルを実行ファイル形式に変換し、制御コードをデコードメモリ群21～24に書き込む過程を概略的に示すフローチャートであり、また、図3は図2のフローチャートに従って生成される実行ファイル200、命令名-命令コード対応テーブル300、デコードメモリテーブル（デコード情報）500等の一例を示す説明図であり、以下、これらの図に従って、この実施の形態1による並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサの動作について説明する。

【0031】この実施の形態1による並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサによれば、以下に示すように、アセンブラソースファイル100が必要とする命令101～106に対応する制御コードだけを選び出し、それらをデコードメモリ群21～24に割り当てる。まず、パス2に従って、以下で示す命令コード逐次割当アセンブラが、一般のアセンブラと同様に、ステップST1において、アセンブラソースファイル100から実行ファイル200を生成して、命令メモリ12に実行ファイル200を格納する。実行ファイル200を生成する場合、アセンブラソースファイル100中に出現した命令名111～114に順番に命令コード201～204を割り当てていく。図3に示す例では、最初に出現した命令名111“nop（非実行命令）”に命令コード201“00”を、命令名112“tfr（転送命令）”に命令コード202“01”を、命令名113“mpy（乗算命令）”に命令コード203“02”を、命令名114“sll（シフト命令）”に命令コード204“03”を順番に割り当て、実行ファイル200を作成する。命令コード201～204は固定であり、異なるデコードメモリテーブル500に対しても同一の命令コード201～204が使用され、後で述べるように、デコードメモリ群21～24は異なるデコードメモリテーブル500により再設定され得る。尚、実行ファイル200中で命令コード201～204の右方の下線の無い部分は、レジスタの指定等に使われるオペランドである。

【0032】次に、パス3を介してデコードメモリ群21～24に格納するデコードメモリテーブル500を作成しデコードメモリ群21～24に設定する過程について説明する。まず、図2に示すように、ステップST2において、命令名111～114と上記の過程において命令名111～114に対して割り当てられた命令コード201～204との対応を示す命令名-命令コード対

応テーブル300を作成する。尚、この命令名-命令コード対応テーブル300の作成は、ステップST1の実行ファイル200の作成過程中に同時に実行され得る。

【0033】次に、ステップST3において、予め定めた、並列処理プロセッサ600上で物理的に実現可能な全ての制御コードと命令名との対応テーブル、即ち仮想命令セット400を使い、さらに、命令名-命令コード対応テーブル300を参照して、命令コードと制御コードとの対応関係を示す対応テーブルであるデコードメモリテーブル500を作成する。例えば、命令コード201“00”にはnopの制御コード401を、命令コード202“01”にはtfrの制御コード402を、命令コード203“02”にはmpyの制御コード403を、命令コード204“03”にはsllの制御コード404を割り当てる。そして、ステップST4において、この結果を用いて並列処理プロセッサ600上のデコードメモリ群21～24を初期化する。即ち、命令コード201～204をアドレス信号（アドレスポート25～28の入力）としてアドレス指定ポート37を介してデコードメモリ群21～24のアドレスポート25～28に入力させ、対応する制御コード401～404をデータ書き込みポート39およびデータ入力ポート29～32を介してデコードメモリ群21～24に書き込む。この際、制御部20は、マルチプレクサ38を制御して、デコードメモリ群21～24のアドレスポート25～28を順次アドレス指定ポート37に接続する。

【0034】この結果、並列処理プロセッサ600の動作時に、実行ファイル200内の命令コード201～204に対応した制御コード401～404が第1～第4の演算部50～53に与えられ、従来例の場合と同様に正しい命令実行が行なわれる。

【0035】次に、図2および図3に示したデコードメモリテーブル500を作成しデコードメモリ群21～24を初期化する過程を実現するために不可欠なソフトウェアツールの構成とその機能とについて説明する。図4はそのソフトウェアツールの構成を示すブロック図であり、図において、4は命令コードを出現順に逐次割り当てることができる命令コード逐次割当アセンブラ、5は制御コードジェネレータである。

【0036】命令コード逐次割当アセンブラ4は、命令名に対して固定された命令コードを割り当てる従来方式のアセンブラとは異なり、図3に示したような、入力されたアセンブラソースファイル100の中に現れる命令101～106に対し、その都度順番に命令コード201～204を割り当てていく。命令コード逐次割当アセンブラ4は、その結果を命令コード201～204の並びとして実行ファイル200に書き出す。他方、命令コード逐次割当アセンブラ4はどの命令名に対してどの命令コードを割り当てたかを、命令名と命令コードとの対応関係を示す命令名-命令コード対応テーブル300に

出力する。制御コードジェネレータ5は、この命令名-命令コード対応テーブル300と並列処理プロセッサ600上で物理的に実現可能な全ての制御コードと命令名との対応関係を示す仮想命令セット400とを元に、命令コードに対応する制御コードを見付けデコードメモリテーブル500へ出力する。そして、上記したように、このデコードメモリテーブル500の内容は命令の起動に先立って、デコードメモリ群21~24へ転送される。

【0037】デコードメモリ群21~24を一度だけ設定する場合について述べてきたが、分割アセンブルやライブラリ参照によって生成された実行ファイル200においては、異なるデコードメモリテーブルが必要となる異なる他の命令コード群が存在する。従って、この場合、制御部20は、異なるデコードメモリテーブルが必要となる異なる他の命令群との境界において、命令メモリ12に書き込まれた再設定命令に従って、デコードメモリ群21~24の再設定を外部に要求するために再設定レジスタ60の内容を例えば“1”に遷移させ、再設定要求出力ポート61を介して再設定要求を外部に出力する。並列処理プロセッサ600の外部に設けられた集積回路等のハードウェアを用いて、この要求に応答して、予め作成しておいた他のデコードメモリテーブル500をデータ書き込みポート39およびデータ入力ポート29~32を介してデコードメモリ群21~24に書き込む。

【0038】以上のように、この実施の形態1によれば、命令語長またはアドレス幅によって決まる命令数よりも多くの命令を仮想的に実現できる効果が得られる。例えば、近藤よしかず等による“ファーストコンバージェンスを示す1.2-GFLOPS ニューラルネットワークチップ(A 1.2-GFLOPS Neural Network Chip Exhibiting Fast Convergence”、国際ソリッドステートサーキットカンファレンステクニカルペーパーダイジェスト(International Solid-State Conference Digest of Technical Papers)、218~219ページ、1994に開示されたSIMD型並列処理プロセッサでは、命令コードフィールドが8bitであり、256種類の命令のみしか実現できなかったものが、この実施の形態1による並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサを採用すれば300~500種類程度の命令まで拡張可能であると期待される。また、命令語長を増大させる必要がないので、実質的に命令語長を減らすことができ、従って、I/Oバンド幅の制限を緩和して1つのシリコン基板上に搭載できる機能を増やすことが可能となる。

【0039】実施の形態2. 図5はこの発明の実施の形

態2による並列処理プロセッサにおける命令数拡張方法を実現する並列処理プロセッサの主要部の構成を示すブロック図であり、図において、図1と同一の部分には同一符号を付して重複説明を省略する。また、図において、40はデコードメモリ群21~24のいずれの内容を書き換えるかを定めるために、書き換え対象命令(即ち、書き換え対象となる命令コード)を指定する命令選択レジスタ、41は書き換え内容(制御コード)を指定する命令定義レジスタである。

【0040】上記実施の形態1による並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサでは、図1に示す制御部20は、図4に示すソフトウェアツール構成図から明かなように、1つのアセンブラソースファイル100に対して、実行ファイル200とデコードメモリテーブル500という2種類のデータを管理しなければならない。また、分割アセンブルやライブラリ参照によって作成した実行ファイル200に対しては、異なるデコードメモリテーブル500が必要となる異なる他の命令群との境界においてデコードメモリ群21~24の書き換え要求を並列処理プロセッサ600の外へ送付し、デコードメモリ群21~24の書き換えが終わるまでは並列処理プロセッサ600の動作を停止させる必要がある。このような動作は、多数のライブラリ関数を参照する大規模なプログラムにおいて、実行速度を低下させる。

【0041】この実施の形態2による並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサは、このような問題に対処するために、デコードメモリ群21~24を書き換える命令およびハードウェアを設け、実行ファイル200中でデコードメモリ群21~24の書き換え処理を完結させるものである。そのために、並列処理プロセッサ600は、実施の形態1による並列処理プロセッサ600の構成に加えて、書き換え対象命令を指定する命令選択レジスタ40と、書き換え内容(制御コード)を指定する命令定義レジスタ41とを備えた。また、実行ファイル200中に含まれ、命令メモリ12に格納された書き換え命令は、デコードメモリ群21~24に新たに書き込まれる書き換え用のデコードメモリテーブル500を含んでいる。

【0042】次に動作について説明する。命令選択レジスタ40および命令定義レジスタ41によって、命令メモリ12中に予め定義しておいたデコードメモリテーブル500の内容をデコードメモリ群21~24へ転送する機能を実現することができる。即ち、以下の3つの命令、命令メモリ12から命令選択レジスタ40へのデコードメモリ群21~24の書き換え対象命令を示すアドレス値を転送する命令、命令メモリ12から命令定義レジスタ41へ新たな制御コードを転送する命令、および、デコードメモリ書き込み命令(命令選択レジスタ40で指定されるアドレス値を持つデコードメモリ群21

～24のいずれか一つの内容を命令定義レジスタ41の内容で書き換える)を用意し、これらの命令を順に繰返すことで命令メモリ12上のデコードメモリテーブル500の内容をデコードメモリ群21～24に転送し書き込むことができる。

【0043】以上のように、この実施の形態2によれば、多数のライブラリ関数を参照する大規模なプログラムにおいても、実効速度を低下させることなく、低頻度命令の制御コードを入れ替えることにより、命令語長またはアドレス幅によって決まる命令数よりも多くの命令を仮想的に実現できる効果が得られる。

【0044】実施の形態3。図6はこの発明の実施の形態3による並列処理プロセッサにおける命令数拡張方法を実現する並列処理プロセッサの主要部の構成を示すブロック図であり、図において、図1と同一の部分には同一符号を付して重複説明を省略する。また、図において、42は高頻度命令の制御コードを記憶するための初期値ROM(読み出し専用メモリ)、43は初期値ROM42中に記憶された制御コードのアドレスおよびデコードメモリ群21～24のアドレスを指定するためのカウンタである。

【0045】通常のLSIにおける内部のデータ転送速度は、負荷容量等の差により、LSI外部に比べて数倍高速である。従って、単一LSIで構成される並列処理プロセッサ600上に、使用頻度の高い制御コード(例えばデータ転送命令や、上記実施の形態2で述べたデコードメモリ書き込み命令など)を部分的に初期化する機能を搭載することにより、外部からの制御コード転送量を削減し、命令起動を高速化することが可能になる。この実施の形態3による並列処理プロセッサにおける命令数拡張方法および並列処理プロセッサでは、かかる制御コード初期化機能を初期値ROM42およびカウンタ43により達成するものである。

【0046】次に動作について説明する。並列処理プロセッサ600の初期化時に1度だけ、カウンタ43を増加させながら、デコードメモリ群21～24へ初期値ROM42に格納された制御コードの初期値を転送し、デコードメモリ群21～24を部分的に初期化する。そして、上記実施の形態1と同様な方法を用いて低頻度命令の制御コードをデコードメモリ群21～24の残りの部分に書き込む。そして、必要ならば、デコードメモリ群21～24に残りの部分の再設定を要求するために再設定レジスタ60の内容を例えば状態“1”に遷移させる。また、デコードメモリ群21～24を部分的に書き換えるのではなく、全てを書き換えるようにしてもよい。しかしながら、この場合には、高頻度の命令に対応した制御コードを優先的にデコードメモリ群21～24に高速に割り当てることが難しくなる。

【0047】以上のように、この実施の形態3によれば、命令語長またはアドレス幅によって決まる命令数よ

りも多くの命令を仮想的に実現できる効果が得られるとともに、外部からの制御コード転送を低頻度命令のみに限定し、高頻度命令をより高速にデコードメモリ群21～24に設定できる効果が得られる。

【0048】尚、上記実施の形態2による命令選択レジスタ40および命令定義レジスタ41を再設定レジスタ60の代わりに設けてもよい。この場合、上記実施の形態2と同様な効果が得られることは言うまでもない。

【0049】また、上記実施の形態1から実施の形態3では、並列処理プロセッサ600はデコードメモリ群21～24、第1から第4の演算部50～53を備えていたが、デコードメモリセル、演算部の数はこれに限定されるものではないことは言うまでもない。

【0050】

【発明の効果】以上のように、請求項1記載の発明によれば、並列処理プロセッサにおける命令数拡張方法では、書き換え可能な素子で構成された複数の命令解読器を設け、ソースファイルのアセンブル時に作成したデコード情報に基づいて複数の命令に対応した複数の制御コードを複数の命令解読器にそれぞれ書き込むように構成したので、命令語長またはアドレス幅によって決まる命令数よりも多くの命令を仮想的に実現できる効果がある。さらに、実質的に命令語長を減らすことができるので、I/Oバンド幅の制限を緩和して1つのシリコン基板上に搭載できる機能を増やすことができる効果がある。

【0051】請求項2記載の発明によれば、並列処理プロセッサにおける命令数拡張方法では、複数の命令と複数の制御コードとの対応関係を示す仮想命令セットと命令名-命令コード対応テーブルとを参照し、複数の命令コードと複数の制御コードとを対応させたデコードメモリテーブルを作成して複数の命令解読器を設定するように構成したので、命令語長またはアドレス幅によって決まる命令数よりも多くの命令を仮想的に実現できる効果がある。

【0052】請求項3記載の発明によれば、並列処理プロセッサにおける命令数拡張方法では、所定の複数の命令コードのそれぞれをアドレスとして指定して命令解読器に書き込むように構成したので、簡易に命令解読器に制御コードを設定でき、命令語長またはアドレス幅によって決まる命令数よりも多くの命令を仮想的に実現できる効果がある。

【0053】請求項4記載の発明によれば、並列処理プロセッサにおける命令数拡張方法では、複数の命令解読器に書き込まれた内容を再設定するための再設定命令を設け、再設定命令を実行した場合に、予め作成した、複数の命令コードと他の複数の制御コードとを対応させた他のデコードメモリテーブルに基づき他の複数の制御コードを複数の命令解読器にそれぞれ書き込むように構成したので、命令語長またはアドレス幅によって決まる命令数よりも多くの命令を仮想的に実現できるうえに、分

割アセンブルやライブラリ参照によって作成した実行ファイル中の異なる他の命令群に応じて命令解読器の内容を書き換えることができる効果がある。

【0054】請求項5記載の発明によれば、並列処理プロセッサにおける命令数拡張方法では、複数の命令解読器に書き込まれた内容を書き換えるための書き換え命令を設け、書き換え命令を実行した場合に予め並列処理プロセッサ内に記憶していた複数の制御コードを複数の命令解読器にそれぞれ書き込むように構成したので、命令語長またはアドレス幅によって決まる命令数よりも多くの命令を仮想的に実現できるうえに、並列処理プロセッサを停止させることなく、分割アセンブルやライブラリ参照によって作成した実行ファイル中の異なる他の命令群に応じて命令解読器の内容を書き換えることができる効果がある。

【0055】請求項6記載の発明によれば、並列処理プロセッサにおける命令数拡張方法では、並列処理プロセッサの初期化の際に、複数の命令解読器の少なくとも一部に一つまたは複数の所定の制御コードを書き込むように構成したので、例えば、制御コードの命令解読器への転送を低頻度命令のみに限定し、特定の命令に関する制御コードをより高速に命令解読器に設定できる効果がある。

【0056】請求項7記載の発明によれば、並列処理プロセッサにおける命令数拡張方法では、上記の所定の制御コードは出現頻度が高い命令に関するものであるように構成したので、例えば、制御コードの命令解読器への転送を低頻度命令のみに限定し、高頻度命令に関する制御コードをより高速に命令解読器に設定できる効果がある。

【0057】請求項8記載の発明によれば、並列処理プロセッサを書き換え可能な素子で構成された複数の命令解読器と、外部から複数の命令解読器のいずれかを選択するために入力されるアドレス信号を受信する第1の入力ポートと、デコード情報に基づき外部から複数の命令解読器に制御コードをそれぞれ書き込むために制御コードを受信する第2の入力ポートとを備えるように構成したので、命令語長またはアドレス幅によって決まる命令数よりも多くの命令を仮想的に実現できる効果がある。さらに、実質的に命令語長を減らすことができるので、I/Oバンド幅の制限を緩和して1つのシリコン基板に搭載できる機能を増やすことができる効果がある。

【0058】請求項9記載の発明によれば、並列処理プロセッサを複数の命令解読器が、第1の入力ポートを介してアドレス信号を受信するためのアドレスポートと、第2の入力ポートを介して制御コードを受信するための複数のデータ入力ポートと、対応する演算器に制御コードを送信するための複数のデータ出力ポートとを備えたランダムアクセスメモリであるように構成したので、命令語長またはアドレス幅によって決まる命令数よりも多

くの命令を仮想的に実現できる効果がある。

【0059】請求項10記載の発明によれば、並列処理プロセッサを複数の命令解読器を再設定すべく書き換えるために再設定要求を外部に出力するための再設定レジスタを備えるように構成したので、分割アセンブルやライブラリ参照によって作成した実行ファイル中の異なる他の命令群に応じて命令解読器の内容を書き換えることができる効果がある。

【0060】請求項11記載の発明によれば、並列処理プロセッサを所定の書き換え命令に応じて複数の命令解読器のいずれか一つを選択するためのアドレスを保持する命令選択レジスタと、書き換え命令に基づき予め保持していた制御コードを命令選択レジスタで指定される複数の命令解読器の一つに書き込むために一旦記憶する命令定義レジスタとを備えるように構成したので、並列処理プロセッサの動作を停止させることなく、分割アセンブルやライブラリ参照によって作成した実行ファイル中の異なる他の命令群に応じて命令解読器の内容を書き換えることができる効果がある。

【0061】請求項12記載の発明によれば、並列処理プロセッサを並列処理プロセッサの初期化の際に複数の命令解読器の少なくとも一部に一つまたは複数の所定の制御コードを書き込むために、所定の制御コードを記憶するための読み出し専用メモリを備えるように構成したので、例えば、制御コードの命令解読器への転送を低頻度命令のみに限定し、特定の命令についての制御コードをより高速に命令解読器に設定できる効果がある。

【図面の簡単な説明】

【図1】 この発明の実施の形態1による並列処理プロセッサの命令数拡大方法を実現するための並列処理プロセッサの主要部を示すブロック図である。

【図2】 この発明の実施の形態1による並列処理プロセッサの命令数拡大方法の処理の概要を示すフローチャートである。

【図3】 この発明の実施の形態1による並列処理プロセッサの命令数拡大方法の処理の過程に作成される実行ファイル、命令名-命令コード対応テーブル、およびデコードメモリテーブル等の一例を示す説明図である。

【図4】 この発明の実施の形態1による並列処理プロセッサの命令数拡大方法の処理を実現するためのソフトウェアツールの構成を示すブロック図である。

【図5】 この発明の実施の形態2による並列処理プロセッサの命令数拡大方法を実現するための並列処理プロセッサの主要部を示すブロック図である。

【図6】 この発明の実施の形態3による並列処理プロセッサの命令数拡大方法を実現するための並列処理プロセッサの主要部を示すブロック図である。

【図7】 従来の並列処理プロセッサの主要部を示すブロック図である。

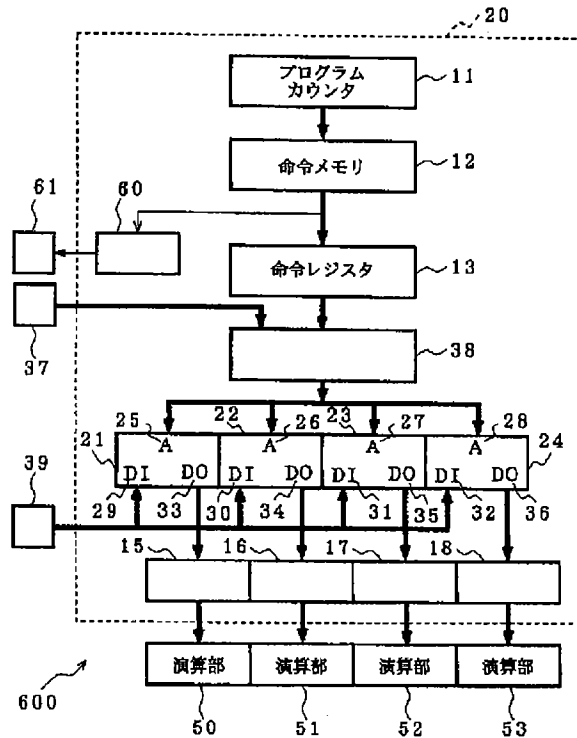
【図8】 SIMD型並列処理プロセッサにおける命令

出現頻度の実測値を示す円グラフ図である。

【符号の説明】

21～24 デコードメモリ群（命令解読器）、25～28 アドレスポート、29～32 データ入力ポート、33～36 データ出力ポート、37 アドレス指定ポート（第1の入力ポート）、39 データ書き込みポート（第2の入力ポート）、40 命令選択レジス

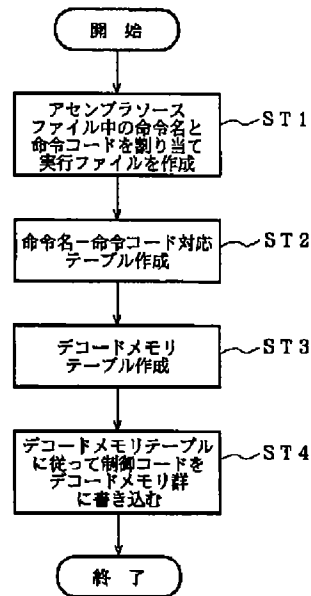
【図1】



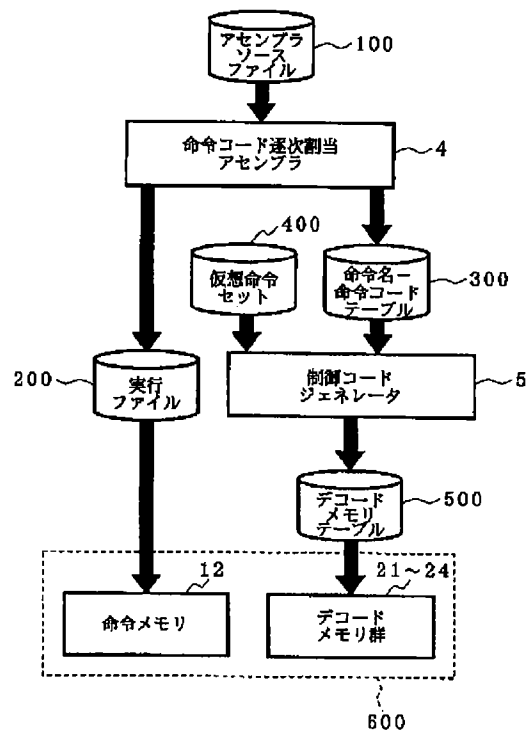
21～24：デコードメモリ群（命令解読器）
25～28：アドレスポート
29～32：データ入力ポート
33～36：データ出力ポート
37：アドレス指定ポート（第1の入力ポート）
39：データ書き込みポート（第2の入力ポート）
80：再設定レジスタ
600：並列処理プロセッサ

タ、41 命令定義レジスタ、42 初期値ROM（読み出し専用メモリ）、50～53 演算部、60 再設定レジスタ、300 命令名-命令コード対応テーブル、400 仮想命令セット、500デコードメモリテーブル（デコード情報）、600 並列処理プロセッサ。

【図2】



【図4】



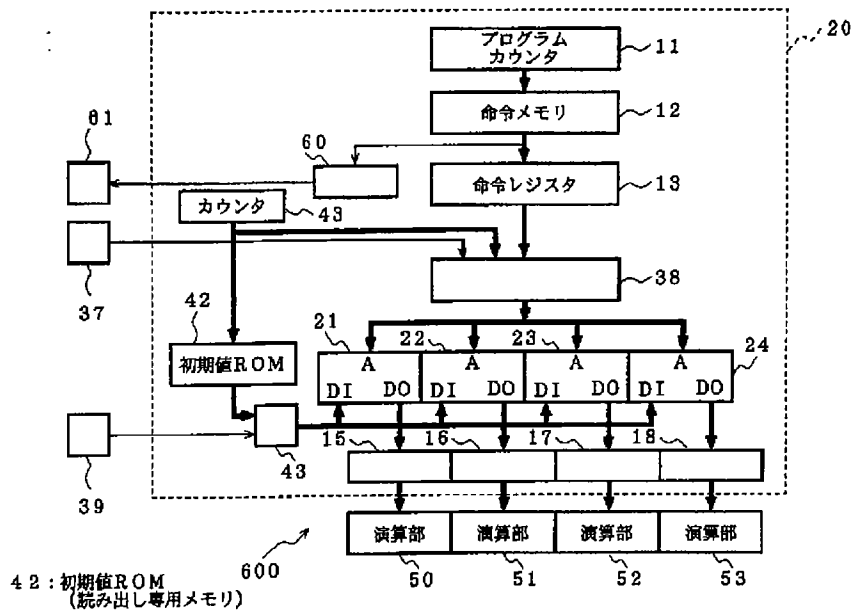
[illegible]

300:命令名-命令コード対応テーブル 400:仮想命令セット 500:デコードメモリテーブル(デコード情報)

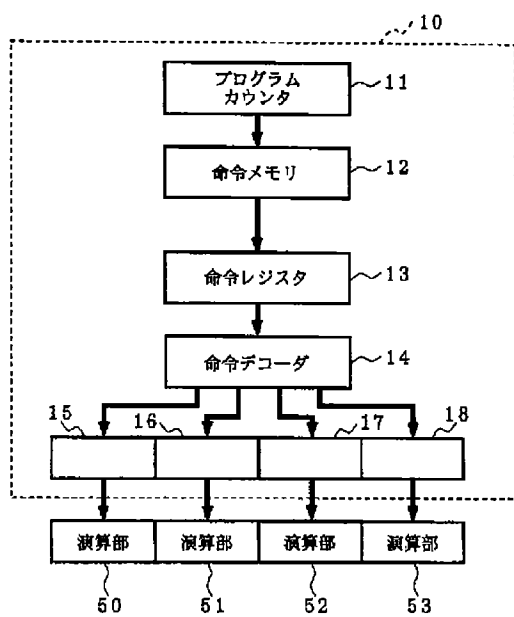
Figure 1 is a block diagram of a computer system 200. The system includes a Program Counter (11), Command Memory (12), Command Register (13), Command Address Register (41), and Command Selection Register (40). These are connected to a central processing unit (38). The central processing unit is connected to a multi-ported memory (21) with four banks (22, 23, 24) and four ports (A, B, C, D). The memory is connected to a bus (15) which is connected to four ALU units (50, 51, 52, 53). The system also includes a Command Selection Register (39) and a Command Selection Register (37).

40: 命令選択レジスタ
41: 命令定義レジスタ

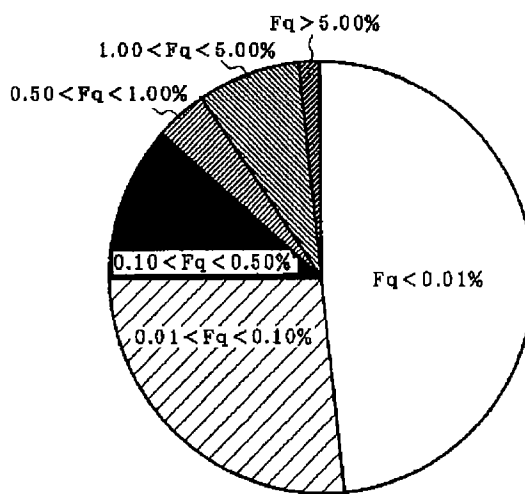
【図6】



【図7】



【図8】



Fq: 命令出現頻度